

支持随机服务请求的云虚拟机按需物理资源分配方法^{*}

曹洁^{1,2,3}, 曾国荪^{1,3}, 匡桂娟^{1,3}, 张建伟², 马海英⁴, 胡克坤^{1,3}, 钮俊⁵



¹(同济大学 计算机科学与技术系, 上海 200092)

²(郑州轻工业学院 软件学院, 河南 郑州 450002)

³(国家高性能计算机工程技术中心同济分中心, 上海 200092)

⁴(南通大学 计算机科学与技术学院, 江苏 南通 226019)

⁵(宁波大学 计算机科学与技术系, 浙江 宁波 315211)

通讯作者: 曹洁, E-mail: cjjiecao@126.com

摘要: 针对云平台按负载峰值需求配置处理机资源、提供单一的服务应用和资源需求动态变化导致资源利用率低下的问题, 采用云虚拟机中心来同时提供多种服务应用. 利用灰色波形预测算法对未来时间段内到达虚拟机的服务请求量进行预测, 给出兼顾资源需求和服务优先等级的虚拟机服务效用函数, 以最大化物理机的服务效用值为目标, 为物理机内的各虚拟机动态配置物理资源. 通过同类虚拟机间的全局负载均衡和多次物理机内各虚拟机的物理资源再分配, 进一步增加服务请求量较大的相应类型的虚拟机的物理资源分配量. 最后, 给出了虚拟机中心基于灰色波形预测的按需资源分配算法 ODRGWF. 模拟实验结果表明, 该算法能够有效地提高云平台中处理机的资源利用率, 对提高用户请求完成率以及服务质量都具有实际意义.

关键词: 云计算; 虚拟化; 随机服务请求; 灰色波形预测; 按需资源分配

中图法分类号: TP316

中文引用格式: 曹洁, 曾国荪, 匡桂娟, 张建伟, 马海英, 胡克坤, 钮俊. 支持随机服务请求的云虚拟机按需物理资源分配方法. 软件学报, 2017, 28(2): 457–472. <http://www.jos.org.cn/1000-9825/5054.htm>

英文引用格式: Cao J, Zeng GS, Kuang GJ, Zhang JW, Ma HY, Hu KK, Niu J. On-Demand physical resource allocation method for cloud virtual machine to support random service requests. Ruan Jian Xue Bao/Journal of Software, 2017, 28(2): 457–472 (in Chinese). <http://www.jos.org.cn/1000-9825/5054.htm>

On-Demand Physical Resource Allocation Method for Cloud Virtual Machine to Support Random Service Requests

CAO Jie^{1,2,3}, ZENG Guo-Sun^{1,3}, KUANG Gui-Juan^{1,3}, ZHANG Jian-Wei², MA Hai-Ying⁴, HU Ke-Kun^{1,3}, NIU Jun⁵

¹(Department of Computer Science and Technology, Tongji University, Shanghai 200092, China)

²(Software College, Zhengzhou Institute of Light Industry, Zhengzhou 450002, China)

³(Tongji Branch, National Engineering and Technology Center of High Performance Computer, Shanghai 200092, China)

* 基金项目: 国家高技术研究发展计划(863)(2009AA012201); 国家自然科学基金(61402244); 上海市优秀学科带头人计划(10XD1404400); 华为创新研究计划(IRP-2013-12-03); 高效能服务器和存储技术国家重点实验室开放基金(2014 HSSA10); 河南省科技创新人才计划([2015]4); 浙江省公益技术应用研究项目(2014C31059)

Foundation item: National High-Tech R&D Program of China (2009AA012201); National Natural Science Foundation of China (61402244); Program of Shanghai Subject Chief Scientist (10XD1404400); Huawei Innovation Research Project (IRP-2013-12-03); Open Foundation of the State Key Laboratory of High-End Server and Storage Technology (2014HSSA10); He'nan Scientific and Technological Innovation Project ([2015]4); Zhejiang Provincial Public Technology Application Research Project (2014C31059)

收稿时间: 2014-09-29; 修改时间: 2015-04-08, 2015-10-20, 2015-12-22; 采用时间: 2016-02-19; jos 在线出版时间: 2016-05-03

CNKI 网络优先出版: 2016-05-04 08:44:06, <http://www.cnki.net/kcms/detail/11.2560.TP.20160504.0844.002.html>

⁴(School of Computer Science and Technology, Nantong University, Nantong 226019, China)

⁵(Department of Computer Science and Technology, Ningbo University, Ningbo 315211, China)

Abstract: Low resource utilization is becoming much more serious in cloud platform which allocates processor resources according to the peak load while providing single service application and facing dynamic variation of resource demand. To address the problem, this study uses cloud virtual machine (VM) center to provide a variety of reasonable service applications simultaneously. Gray wave forecasting algorithm is adopted to predict the future load of service requests and a VM service utility function is proposed by taking resource requirements and service priorities into account. Each VM inside a physical machine dynamically configures physical resources to maximize the service utility value of the physical machine. Besides, by applying the global load balancing and multi-time physical resource redistribution for each virtual machine in the same physical machine, the number of physical resources assigned to the VMs whose service request amount is much larger is further increased. In the end, on-demand resource reconfiguration algorithm ODRGWF based on grey wave forecasting is put forward. The simulation results show that the proposed algorithm can effectively improve processor resource utilization, which is of practical significance to improve user request completion rate and service quality.

Key words: cloud computing; virtualization; random service request; grey wave forecasting; on-demand resource allocation

云计算以网络化的方式组织和聚合计算与通信资源,以虚拟化的方式为用户提供可以缩减或扩展规模的计算资源.虚拟化软件如 VMWare, Xen^[1,2]和 KVM,为云计算按需资源分配提供了强大的技术支持.通过虚拟化软件,可将一台物理机实例化为多个虚拟机,将多台物理机的剩余计算资源虚拟化成一台虚拟机.由此,云中心的体系结构通常分为 3 层^[3]:底层是大量的物理资源;中间层是虚拟化服务层,由大量的虚拟机实例组成;最高层是云计算应用层,不同类型的应用有不同的服务质量需求.

目前,一个大型的云中心往往托管了数以万计的 x86 服务器,出于对安全、可靠性和性能的考虑,这些服务器基本上只运行一种应用服务.通常云中心的服务器数量是根据峰值服务请求进行配置的,在大多数时间段内,大多数服务器处于空闲状态,造成服务器利用率低下.据统计,企业数据中心的平均资源利用率通常低于 15% ~ 20%.此外,不同云中心在不同时间段的繁忙程度可能差别很大,在某时间段,一些云中心的负载比较重,而其他云中心的负载可能比较轻.如果一个服务器能够同时提供多种服务应用,原来由多个云中心分别提供服务就可以用 1 个云中心来同时提供这些服务.虚拟化技术为这个设想的实现提供了可行的解决方案,通过服务器虚拟化,一个物理服务器可被虚拟化成若干个虚拟服务器使用.不同的虚拟服务器也称为虚拟机(virtual machine,简称 VM).提供资源虚拟化技术的软件称为虚拟机监视器(virtual machine monitor,简称 VMM).虚拟机监视器直接运行在硬件上或操作系统中,虚拟机运行在虚拟机监视器上.虚拟机操作系统使用的是虚拟化资源,虚拟机监视器负责虚拟资源到物理资源的映射.这样,云中心可根据不同类型服务应用负载的不同,减少执行轻载服务应用的虚拟机的物理资源,增加执行重载服务应用的虚拟机的物理资源.

虚拟化削弱了操作系统和硬件之间的相互依赖,已经成为资源动态配置的一种有效手段.通常,在云中心有 3 种级别的资源管理方法:(1) 应用级资源管理.属于分布式资源管理方法,由虚拟机本身负责虚拟机之间资源分配的有效性和公平性.(2) 主机级资源管理.属于集中式资源管理方法,由主机内核或虚拟机监视器向该主机上的虚拟机分配资源.(3) 集群级资源管理.属于集中式资源管理方法,由控制中心根据实际工作负载,调整虚拟节点间的资源分配.

本文所提出的支持随机服务请求的云虚拟机按需物理资源分配方法 ODRGWF 属于主机级资源管理方法.该分配方法通过实时跟踪物理机内各虚拟机应用负载的变化,及时调整物理资源在虚拟机间的资源配置,缓解发生在物理机内的各虚拟机之间的资源竞争.此外,通过服务请求全局负载均衡缓解服务请求发生在物理机之间的资源竞争.

1 相关工作

云服务提供商能否赢利,取决于其所提供的云平台的稳定性、云平台的资源利用率以及虚拟机的价格等因素.如何根据应用、服务负载的变化为其所在的虚拟机及时、有效地分配物理资源,保证既不会因为资源缺乏而影响业务系统运行,也不会造成严重的资源浪费,已成为提高云虚拟机中心资源利用率和服务质量的关键.

关键需求.当前云计算环境下虚拟机资源分配的研究主要集中在以下几个方面:(1) 基础设施,即服务级的虚拟机分配机制的研究,即根据用户对虚拟机资源需求的动态变化,实时地为虚拟机分配资源.(2) 基于优先级的资源分配.云服务提供商将云用户的服务请求按照优先级进行分类,优先为执行高优先级服务请求的虚拟机分配资源,然后为执行低优先级服务请求的虚拟机分配资源,从而最大化云服务提供商的收益.(3) 基于服务请求量预测的资源分配.在这种方式中,云服务提供商首先对云用户的服务请求量进行预测,然后按照所预测的服务请求量为相应的云用户配置虚拟机资源.这种方式由于对用户的服务请求量进行了预测,因而在保障用户服务质量的同时可以提高资源的利用率.

当前,为虚拟机按需分配资源的问题已得到国内外学者的广泛研究.在国外,VMWare DRS^[4]通过在物理机之间迁移虚拟机为虚拟机实现按需资源分配.Padala 等人^[5]提出了一种自适应资源控制系统,利用虚拟化技术动态调整 CPU 和存储资源在各应用层的资源份额以满足应用级的服务质量.Zhao 等人^[6]引入内存均衡器(MEB)动态监控每个虚拟机的内存使用情况,预测虚拟机的内存需求并周期性地虚拟机之间进行内存再分配.Wang 等人^[7]以集中的方式在多层应用间优化全局资源分配.Song 等人^[8,9]在基于虚拟机的数据中心,通过物理机内虚拟机的动态资源再分配来实现资源分配的优化.在国内,吴恒等人^[10]以收益率最大化作为资源重配置策略选择的依据,提出了一种收益敏感的资源按需提供方法.米海波等人^[11]面向 Web 应用,根据不断变化的资源需求以在线方式重配置集群,实时地确定集群当前运行的节点数量及其上部署的虚拟机类型.许力等人^[12]将云计算中的虚拟资源分配问题建模为多目标优化模型,针对不同特征的虚拟主机和服务器需求生成虚拟资源分配方案.文雨等人^[13]提出了一种面向应用服务级目标的虚拟化资源管理方法,通过动态调整虚拟机资源分配来实现每个应用的服务器目标,通过仲裁不同应用的资源分配请求来控制虚拟机在非虚拟化资源上的竞争干扰.师雪霖等人^[14]提出了云效用最大化(cloud utility maximization,简称 CUM)资源分配模型,以达到效用最大为调度目标,通过求解 CUM 优化问题得到最优的虚拟机和物理机映射关系.金海等人^[15]提出了资源使用状态驱动的资源再配置方法,自动适应动态负载变化来满足应用性能的资源需求.杨雷等人^[16]提出了一种考虑虚拟机间性能互扰的虚拟资源分配方法,给出了基于虚拟机负载模式的性能互扰度预测方法,将虚拟机间性能互扰的预测结果作为虚拟资源分配的依据.

文献[4]通过虚拟机在物理机之间的迁移来实现虚拟机所需物理资源的按需分配,与在物理机内部对虚拟机进行资源再分配相比,在物理机之间进行虚拟机迁移将花费较高的代价.文献[7]中集中式的全局资源分配不可避免地会带来计算的复杂性、单点故障和资源分配不及时问题.文献[5,6,8,9,15]以最近的负载作为为虚拟机分配资源的依据,没有兼顾到将来的负载变化,易造成资源分配不足或过量.文献[8-14,16]没有考虑全局负载均衡,无法最大程度地充分利用整个系统中的物理资源.针对上述资源分配方法在资源分配中存在的问题,本文用灰色波形预测方法预测每个虚拟机在未来一段时间的负载,将其作为为虚拟机分配资源时虚拟机负载的大小;兼顾不同类型服务请求的优先程度,以最大化物理机内各虚拟机的总服务效用为目标,为虚拟机分配物理机上的资源;通过系统中同类虚拟机间的负载均衡和物理机内各虚拟机间物理资源的多次分配,使得各虚拟机的处理能力与其上的负载大致相匹配.

2 云中心虚拟机物理资源按需分配模型和假设

2.1 云中心虚拟机物理资源按需分配模型

云中心的应用类型可分为 CPU 密集型、数据密集型和 I/O 密集型等.例如,数据库应用是 CPU 密集型,用户数据分析是数据密集型,视频点播应用是 I/O 密集型,不同类型的服务请求对资源的需求强度差别很大.在采用服务器虚拟化之前,3 种不同的应用需要分别运行在 3 个独立的物理服务器(物理机)上;在采用服务器虚拟化之后,这 3 种应用可运行在由一个物理服务器托管的 3 个独立的虚拟服务器(虚拟机)上.服务器虚拟化为虚拟服务器提供了能够支持其运行的硬件资源抽象,包括虚拟 BIOS、虚拟处理器、虚拟内存、虚拟设备与 I/O.

通过虚拟化技术在物理机内虚拟出不同类型的虚拟机来执行相应的应用,既满足了不同类型服务请求对服务的需求,也避免了不同种类型的服务请求对同一资源的竞争,还提高了处理机的资源利用率.在云中心,为执行不同类应用的虚拟机按需分配恰当的物理资源,对保证到达的服务请求的服务质量至关重要.为此,云中心按需资源分配需要处理两类分配:(1) 将服务请求分配到虚拟机;(2) 将物理机的物理资源分配给其上的虚拟机.本文将据此设计两层按需资源分配架构,如图 1 所示.

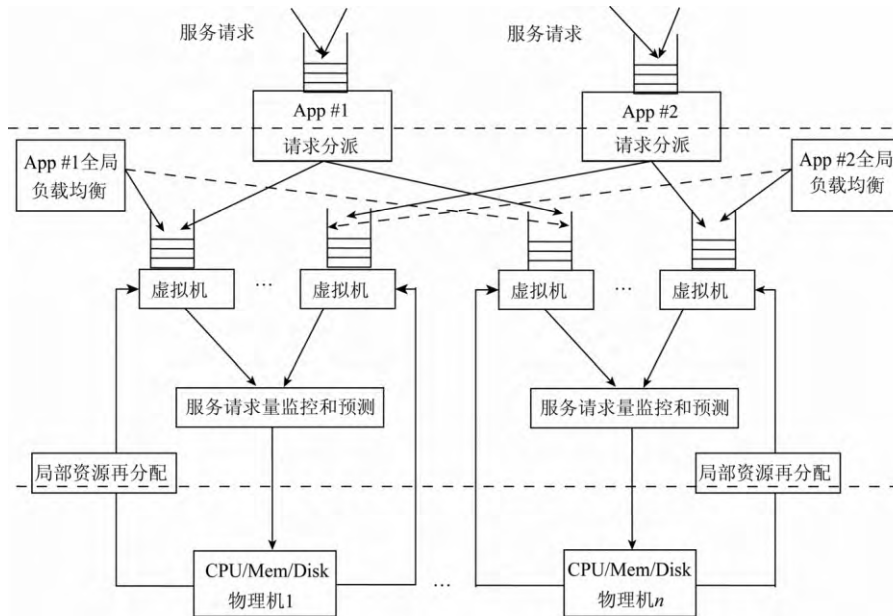


Fig.1 Two-Tiered on-demand resource allocation mechanism

图 1 两层按需资源分配机制

2.2 云中心物理机和虚拟机的描述

定义 1(物理机). 物理机是指在云计算环境中直接参与服务请求处理的计算机,又称计算节点.物理机 pm_i 可用一个二元组表示: $pm_i=(cpu_i, mem_i)$,其中, cpu_i 表示物理机的处理速度大小, mem_i 表示物理机的内存大小.整个云计算系统的物理机集合记为 $PM=(pm_1, pm_2, \dots, pm_n)$, n 表示云计算系统中的物理机个数.

定义 2(虚拟机). 虚拟机是指在物理机上利用虚拟化技术按照应用的需求虚拟出的计算机,用于执行某类型应用服务.物理机 pm_i 上的虚拟机 vm_{ij} 可用一个四元组表示: $vm_{ij}=(cpu_i, mem_i, w_{ij}^{cpu}, w_{ij}^{mem})$,其中, cpu_i 为虚拟机所在物理机的处理速度大小, mem_i 为虚拟机所在物理机的内存大小, $w_{ij}^{cpu} (<1)$ 为虚拟机 vm_{ij} 从物理机的 cpu_i 所分到的 CPU 百分比($\sum_{j=1}^m w_{ij}^{cpu} = 1$, m 为物理机中虚拟机的个数), $w_{ij}^{mem} (<1)$ 为虚拟机 vm_{ij} 从物理机的 mem_i 所分到的内存百分比($\sum_{j=1}^m w_{ij}^{mem} = 1$),因此,虚拟机 vm_{ij} 的实际处理速度为 $w_{ij}^{cpu} cpu_i$,虚拟机 vm_{ij} 的实际内存为 $w_{ij}^{mem} mem_i$.

由于当前的 VMMs 技术还不支持通过一个虚拟机来使用遥远物理机的物理资源,因此,本文只采用一个物理机实例化为多个虚拟机的方式来同时服务多个应用.一个物理机所能运行的虚拟机个数依赖于物理机的性能和虚拟机的大小,物理机运行的虚拟机数目称为物理机的容量.

2.3 虚拟机物理资源按需分配的基本假设

云中心虚拟机物理资源按需分配是指根据最近时间段到达虚拟机的服务请求数量的变化,按某种资源分配策略动态调整分配给虚拟机的物理资源(如 CPU 时间槽的大小、内存的大小等).为了更好地分析问题,我们

做如下假设:每个物理机所运行的不同类型的虚拟机个数都为 m ,一种类型的虚拟机只执行 1 类服务请求;云用户出于偏好,请求某些物理机中的虚拟机处理自己的服务请求,本文暂不考虑服务请求是如何分配到虚拟机上的.为了下文叙述方便,我们对文中所用的主要符号进行归纳.

m : 每个物理机所运行的不同类型的虚拟机数目.

$d_{ij}^{(k)}$: 在第 k 个时间间隔到达虚拟机 vm_{ij} 的服务请求数.

$d_{ij}^{(k)}$: 在第 k 个时间间隔虚拟机 vm_{ij} 完成的服务请求数.

r : 物理机内的虚拟机可用的总 CPU 或其他资源量.

$r_{ij}^{need}(t)$: vm_{ij} 在时间 t 时的资源需求量,正比于虚拟机上的服务请求数.

$r_{ij}^{allo}(t)$: 在时间 t 为 vm_{ij} 分配的资源量.

sw_{ij} : vm_{ij} 的服务权重权重越大, vm_{ij} 上的服务请求等级就越高,其值在系统运行期间不变.

cq_{ij} : vm_{ij} 上的每个服务请求的计算量.

3 随机服务请求数量的灰色波形预测方法

3.1 云服务请求的随机性和规律

在云环境下,用户提交服务请求的时间往往是不确定的,导致任务到达云计算系统是随机的.美国的 Dinda 对到达处理机的负载进行了长期跟踪抽样,揭示了大型分布式系统中处理机负载的变化规律^[18]:(1) 自相关性.自相关性体现了预测的流量与过去的流量统计量有一定的相关性.从时间序列的角度看,自相关性可以认为不同时间尺度上,用户访问量对时间的分布是相似的.(2) 周期性.周期性是指用户访问量的时间序列在时间轴上表现出来的一种周而复始的特性.(3) 混沌性.混沌是指确定的非线性系统在一定条件下所呈现出的不确定或者不可预测的随机现象.对于用户访问负载这样的模型,其混沌性宏观上主要表现为对于细微扰动敏感的蝴蝶效应,在微观上表现为无穷嵌套的几何自相似性.由上述处理机负载的变化规律可知,负载的变化可以看作一种时间序列过程,随时间变化有很强的关联性.基于上述负载特性,本文采用灰色波形预测法^[19]对将来到达虚拟机的服务请求数量进行预测.

3.2 未来服务请求数量的灰色波形预测模型

由于到达每个虚拟机的服务请求数量的灰色波形预测过程都相同,我们只对 1 个虚拟机描述灰色波形预测的计算过程.下面首先给出灰色预测模型.

在一个资源分配周期 T 内,连续 n 个等时间间隔内到达虚拟机的服务请求数量序列记为 $X^{(0)}$,即 $X^{(0)}=(x^{(0)}(1), x^{(0)}(2), \dots, x^{(0)}(n))$,其中, $x^{(0)}(i)$ 表示在第 i 个时间间隔内到达虚拟机的服务请求数量.灰色预测过程如下:对刚过去的资源分配周期内的 n 个时间间隔内到达的服务请求数量序列做 1 阶累加运算,设生成的一阶累加序列为 $X^{(1)}=(x^{(1)}(1), x^{(1)}(2), \dots, x^{(1)}(n))$,其中, $x^{(1)}(i) = \sum_{k=1}^i x^{(0)}(k)$, $i=1, 2, \dots, n$.然后,根据观测的时间序列建立灰色模型,即建立微分方程: $\frac{dx^{(1)}}{dt} = ax^{(1)} = b$, 其中,参数 a 称为发展灰数, b 称为内生控制灰数.设 $\hat{a} = \begin{pmatrix} a \\ b \end{pmatrix}$ 为待估计的参数向量,利

用最小二乘估计法求解可得 $\hat{a} = (B^T B)^{-1} B^T Y$, 其中, $B = \begin{pmatrix} -\frac{1}{2}[x^{(1)}(1) + x^{(1)}(2)] & 1 \\ -\frac{1}{2}[x^{(1)}(2) + x^{(1)}(3)] & 1 \\ \vdots & \vdots \\ -\frac{1}{2}[x^{(1)}(n-1) + x^{(1)}(n)] & 1 \end{pmatrix}$, $Y = \begin{pmatrix} x^{(0)}(2) \\ x^{(0)}(3) \\ \vdots \\ x^{(0)}(n) \end{pmatrix}$.求解微分方程,

即可得预测模型: $\hat{x}^{(1)}(k+1) = \left[x^{(0)}(1) - \frac{b}{a} \right] e^{-ak} + \frac{b}{a}, k=1, 2, \dots, n$. 通过累减, 可得原始序列的拟合及预测模型 $\hat{x}^{(0)}(k+1) = \hat{x}^{(1)}(k+1) - \hat{x}^{(1)}(k) = (1 - e^a)(x^{(0)}(1) - \frac{b}{a})e^{-ak}, k=1, 2, \dots, n$, $\hat{x}^{(0)}(k+1)$ 即为预测得到的在第 $k+1$ 个时间槽到达虚拟机的服务请求数.

从一些著名的大型云服务供应商关于服务请求的历史记录可以看出, 服务请求负载波动的幅度比较大, 且具有一定的波形, 因此, 我们用灰色波形预测对虚拟机未来的服务请求负载进行预测, 灰色波形预测实际上是对系统行为变化的波形进行预测. 具体过程如下.

设局部资源再分配周期 T 的 n 个等时间间隔到达的服务请求负载序列为 $X=(x(1), x(2), \dots, x(n))$, 称 $x_k=x(k)+(t-k)[x(k+1)-x(k)]$ 为序列 X 的第 k 段折线图形, $k=1, 2, \dots, n-1$. 称 $\{x_k=x(k)+(t-k)[x(k+1)-x(k)] | k=1, 2, \dots, n-1\}$ 为序列 X 的折线, 仍记为 X , 即 $X=\{x_k=x(k)+(t-k)[x(k+1)-x(k)] | k=1, 2, \dots, n-1\}$. 设 $\sigma_{\max} = \max_{1 \leq k \leq n} \{x(k)\}$, $\sigma_{\min} = \min_{1 \leq k \leq n} \{x(k)\}$, 则有任给 $\xi \in [\sigma_{\min}, \sigma_{\max}]$, 称 $X=\xi$ 为 ξ 的等高线; 称方程组

$$\begin{cases} X = \{x_k = x(k) + (t-k)[x(k+1) - x(k)] | k=1, 2, \dots, n-1\} \\ X = \xi \end{cases}$$

的解 $(t_i, x(t_i)) (i=1, 2, \dots)$ 为 ξ 等高点. 其中, ξ 等高点是折线 X 与 ξ 等高线的交点. 显然, 根据如上设定, 可以得到如下结论, 若 X 的第 i 段折线上有 ξ 等高点, 则其坐标为 $(i + \frac{\xi - x(i)}{x(i+1) - x(i)}, \xi)$. 对等高时刻序列建立 GM(1,1) 模型预测等高时刻的预测值, 进而得到预测的波形.

对于最佳资源再分配周期 T 和 T 的等时间间隔个数 n 的取值, 我们采用文献[17]求解 $\min \left\{ \sum_{i=1}^n \left[\sum_{t=t_{i-1}}^{t_i} (X(t) - r_i)^2 \right] + f(n) \right\}$ 来得到.

4 物理机内虚拟机按需资源分配的效用模型

从图 1 可知, 局部资源再分配用来动态地为物理机内的各虚拟机分配物理资源. 局部资源再分配周期到来时, 物理机 pm_i 内的虚拟机 vm_{ij} 上存在的服务请求数 sn_{ij} 为 $sn_{ij} = \sum_{k=1}^n (a_{ij}^{(k)} - d_{ij}^{(k)})$. 根据刚过去的资源再分配周期内虚拟机 vm_{ij} 上服务请求的到达情况, 预测下一个资源再分配周期单位时间内到达虚拟机 vm_{ij} 的平均服务请求数量 fn_{ij} , 其预测值为 $fn_{ij} = \left(\sum_{k=1}^n \hat{x}_{ij}^{(0)}(n+k) + s_{ij} \right) / T$, 并以此作为按需资源再分配时虚拟机 vm_{ij} 上的服务请求数量的大小. 因此, 在下一个资源再分配周期内, 单位时间内完成到达虚拟机 vm_{ij} 的服务请求所需的物理资源量为 $r_{ij}^{\text{need}} = fn_{ij}cq_{ij}$.

4.1 虚拟机资源分配的服务效用函数

在实际中, 当虚拟机的资源利用率高于一定值时, 随着资源利用率的进一步增加, 该虚拟机上服务请求的平均完成时间将迅速增大, 导致用户服务体验的好感迅速下降. 在云中心, 每个用户都希望自己的服务请求能够得到优质、高效的服务, 即花费少、完成时间短. 通常用户支付的费用越高、完成服务请求所用的时间越短, 云中心获得的服务效用就越大, 服务效用可看作是完成时间和支付费用的函数. 为了简化问题的讨论, 这里将完成时间定为按 SLA 约定的期望完成时间, 至于动态完成时间的服务效用, 可通过边际效用或柯布道格拉斯效用函数来定义. 本文暂不考虑动态完成时间的服务效用取值问题. 令虚拟机 vm_{ij} 完成单位计算量所获得的服务效用为 sr_{ij} , 定义虚拟机 vm_{ij} 服务权重 sw_{ij} 为 $sw_{ij} = sr_{ij} / \sum_{k=1}^m sr_{ik}$, 虚拟机的权重越大, 完成单位服务请求计算量获得的效用就越大.

定义 3(虚拟机服务效用函数). 在为物理机内各虚拟机进行资源再分配时, 假定通过预测得到虚拟机 vm_{ij} 上的平均服务请求数为 fn_{ij} , 虚拟机 vm_{ij} 的服务权重为 sw_{ij} , 若为虚拟机 vm_{ij} 所分配的资源量为 r_{ij}^{allo} , 则虚拟机 vm_{ij}

的服务效用函数 $u_{ij}(fn_{ij}, sw_{ij}, r_{ij}^{allo})$ 定义为

$$u_{ij}(fn_{ij}, sw_{ij}, r_{ij}^{allo}) = 1 - e^{-\frac{r_{ij}^{allo} sw_{ij}}{r_{ij}^{need}}} = 1 - e^{-\frac{r_{ij}^{allo} sw_{ij}}{fn_{ij} c q_{ij}}}.$$

从上述表达式可以看出,为虚拟机分配的资源越多,所得到的效用函数值就越大;在资源需求和资源分配量相同的情况下,虚拟机的服务权重越大,效用函数值就越大,即完成重要的服务请求能够获得更大的服务效用.刚开始,效用的提高是指数级的,表明为虚拟机分配更多资源的重要性,随后继续分配更多的资源只能带来较小的效用提高.在 sw_{ij}/r_{ij}^{need} 取不同数值时,为虚拟机分配相同的资源量,虚拟机所获得的效用值 $u_{ij}(fn_{ij}, sw_{ij}, r_{ij}^{allo})$ 的变化情况如图 2 所示.从图中可以看出,当分配的资源数量从 0 增加到 20 时,上面两条曲线效用增加得比较快,这表明给这两种曲线所代表的虚拟机分配资源能够获得更多的效用;当从 30 增加到 50 时,最下面的曲线的效用比上面两条曲线增加得快,这表明给最下面的曲线所代表的虚拟机分配资源能够获得更多的效用.由图 2 可知,按效用最大化为虚拟机分配物理资源时,上面提出的虚拟机服务效用函数可有效抑制系统过分地给某些虚拟机分配物理资源.即通过上述虚拟机服务效用函数,我们可将物理机的效用最大化问题分解为物理机内的各虚拟机分配多少物理资源,使得虚拟机的效用之和最大的问题.

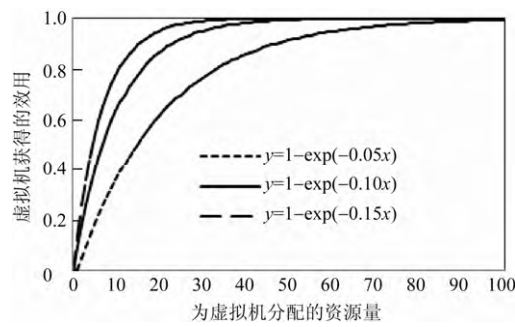


Fig.2 Utility of the VMs

图 2 虚拟机的效用

4.2 最大化物理机内虚拟机总服务效用的物理资源分配求解方法

为物理机内的各虚拟机分配物理资源,最大化整个物理机的服务效用值的问题可以公式化地描述如下:

$$\begin{cases} \max \sum_{j=1}^m u_{ij} = \max \sum_{j=1}^m \left(1 - e^{-\frac{r_{ij}^{allo} sw_{ij}}{r_{ij}^{need}}} \right) = \max \sum_{j=1}^m \left(1 - e^{-\frac{r_{ij}^{allo} sw_{ij}}{fn_{ij} c q_{ij}}} \right). \\ \sum_{j=1}^m r_{ij}^{allo} = r \\ r_{ij}^{allo} \geq 0, j = 1, 2, \dots, m \end{cases}$$

当 $\sum_{j=1}^m r_{ij}^{allo} = r$ 时,求解物理机的效用最大化问题,就是求解物理机内所有虚拟机效用之和最大的问题,即 $\max \sum_{j=1}^m u_{ij}$. 这是一个条件极值问题,可用 Lagrange 乘法法求解.求解过程如下:首先构造 Lagrange 函数

$L(r_{i1}^{allo}, r_{i2}^{allo}, \dots, r_{im}^{allo}, \lambda) = \sum_{j=1}^m \left(1 - e^{-\frac{r_{ij}^{allo} sw_{ij}}{r_{ij}^{need}}} \right) + \lambda \left(\sum_{j=1}^m r_{ij}^{allo} - r \right)$, L 是关于 r_{ij}^{allo}, λ 的函数.然后,根据多元函数取得极值的必要条件,令所有一阶偏导数为 0,故有:

$$\begin{cases} \frac{\partial L}{\partial r_{ij}^{allo}} = 0, j = 1, 2, \dots, m \\ \frac{\partial L}{\partial \lambda} = 0 \end{cases}, \text{ 即 } \begin{cases} \frac{sw_{ij}}{r_{ij}^{need}} e^{-\frac{r_{ij}^{allo} sw_{ij}}{r_{ij}^{need}}} + \lambda = 0, j = 1, 2, \dots, m \\ \sum_{j=1}^m r_{ij}^{allo} - r = 0 \end{cases}.$$

由 $\frac{sw_{ij}}{r_{ij}^{need}} e^{\frac{r_{ij}^{allo} sw_{ij}}{r_{ij}^{need}}} + \lambda = 0$, 得 $r_{ij}^{allo} = \frac{(\ln sw_{ij} - \ln r_{ij}^{need} - \ln(-\lambda)) r_{ij}^{need}}{sw_{ij}}$, 由此式可知, 只要知道 $\ln(-\lambda)$ 的值, 我们就可以求得应该分配给虚拟机 vm_{ij} 的资源量 r_{ij}^{allo} . 将 $r_{ij}^{allo} = \frac{(\ln sw_{ij} - \ln r_{ij}^{need} - \ln(-\lambda)) r_{ij}^{need}}{sw_{ij}}$ 代入 $\sum_{j=1}^m r_{ij}^{allo} - r = 0$, 得 $\ln(-\lambda) = \frac{\sum_{j=1}^m \frac{\ln sw_{ij} - \ln r_{ij}^{need}}{sw_{ij}} - r}{\sum_{j=1}^m \frac{r_{ij}^{need}}{sw_{ij}}}$. 将 $\ln(-\lambda)$ 的值代入前面的 r_{ij}^{allo} 表达式, 即可得到一组解 $(r_{i1}^{allo}, r_{i2}^{allo}, \dots, r_{im}^{allo})$. 由于进行资源分

配所产生的效用值存在最大值, 因此唯一解 $(r_{i1}^{allo}, r_{i2}^{allo}, \dots, r_{im}^{allo})$ 就是使物理机的效用最大时, 各虚拟机应分配的资源量.

5 基于灰色波形预测的云虚拟机中心虚拟机按需物理资源分配

在云虚拟机中心, 为了决定何时进行资源动态调整以及如何调整, 需要对物理机和虚拟机的资源使用情况进行量化监控. 物理机资源 r 的剩余量 r_{sur} 可表示为 $r_{sur} = r_{tot} - \sum_{k=1}^m r_{ik}^{allo} \times ur_{ik}$, 其中, r 为 CPU 或内存, m 为物理机中虚拟机的个数, ur_{ik} 为虚拟机 vm_{ik} 资源 r 的资源利用率, r_{ij}^{allo} 为虚拟机 vm_{ik} 所分配的 r 资源量, r_{tot} 为资源 r 的总量. 剩余量 r_{sur} 这个数值不包括 Hypervisor 使用的资源以及物理机系统使用的资源, 因此, 在使用这个公式时需要排除固定资源使用量, 固定资源使用量可以通过运行零负载的物理机而获得.

5.1 物理机内虚拟机按需资源再配置策略

在局部资源分配周期每个时间间隔内, 监控资源的使用情况, 根据公式 $r_{sur} = r_{tot} - \sum_{k=1}^m r_{ik}^{allo} \times ur_{ik}$ 获取资源 r 的剩余量. 对物理机内的各个虚拟机, 检查其物理资源的实际使用情况, 在时刻 t , 若虚拟机 vm_{ij} 实际使用资源 r 的数量 r_{ij}^{act} 小于分配给它的资源量 r_{ij}^{allo} , 且 $r_{ij}^{allo} - r_{ij}^{act} \geq \varepsilon r_{ij}^{allo}$, 则将这些虚拟机放入可获取资源队列 que_{sur} , 其中, ε 是 $0 < \varepsilon < 1$ 的常数, 用来控制虚拟机资源调整的频度. 若虚拟机 vm_{ij} 上的负载对虚拟机资源 r 的实际需求数量 r_{ij}^{act} 大于分配给它的资源量 r_{ij}^{allo} , 且 $r_{ij}^{act} - r_{ij}^{allo} \geq \delta r_{ij}^{allo}$, δ 的定义类似于 ε , 则将这些虚拟机放入资源需求队列 que_{need} . 如果虚拟机连续两个时间间隔内都满足 $r_{ij}^{act} - r_{ij}^{allo} \geq \delta r_{ij}^{allo}$, 且它们对物理资源的需求不超过 r_{sur} , 则通过虚拟机动态资源调整技术, 将可获取资源队列 que_{sur} 里的虚拟机所多余的部分资源分配给这些虚拟机; 若这些虚拟机对物理资源的需求超过了 r_{sur} , 则通过最大化物理机内虚拟机总服务效用的原则为各虚拟机重新分配物理资源. 若处于资源需求队列 que_{need} 中的虚拟机在下一个时间间隔内 $r_{ij}^{act} - r_{ij}^{allo} < \delta r_{ij}^{allo}$, 则将其从 que_{need} 中移出, 恢复为普通的虚拟机. 对处于可获取资源队列 que_{sur} 中的虚拟机采用类似的处置方法. 之所以延迟虚拟机资源分配, 是为了防止因物理资源配置过于频繁而造成各虚拟机之间物理资源分配的动荡.

5.2 云虚拟机中心基于全局负载均衡的虚拟机按需资源再配置策略

单个物理机内各虚拟机的最优资源分配不一定能使虚拟机中心的虚拟机资源分配最优, 这是因为各个物理机内虚拟机上的服务请求量可能差异很大, 导致不同物理机的负载差异巨大. 因此, 要实现整个虚拟机中心按需资源分配, 需要在各物理机内的同类型的虚拟机之间进行负载均衡, 进而为不同类的服务请求按需提供所需的物理资源. 虽然采用虚拟机迁移的方法也可以实现负载均衡, 但是迁移虚拟机会带来相当大的额外资源消耗, 尤其是网络资源, 而且在迁移的过程中, 应用的性能也会受到影响. 因此, 本文采用迁移服务请求的全局负载均衡的方法, 实现虚拟机中心为服务请求按需再分配资源.

(1) 虚拟机负载的度量

准确掌握云平台中各虚拟机的负载信息是进行负载均衡的基础, 由于虚拟机之间的处理能力可能不同, 因

而不同虚拟机对相同的服务请求计算量所呈现出的负载程度不同.因此,在度量虚拟机的负载时,既要考虑其上服务请求的总计算量,也要考虑到虚拟机的处理能力,下面给出同时反应虚拟机处理能力和虚拟机上服务请求总计算量的虚拟机负载率的定义.

定义 4(虚拟机负载率). 将虚拟机 vm_{ij} 上的服务请求的总计算量 L_{ij} 与虚拟机的处理速度 $w_{ij}^{cpu}cpu_i$ 的比值定义为虚拟机 vm_{ij} 的负载率 lr_{ij} ,即 $lr_{ij} = \frac{L_{ij}}{w_{ij}^{cpu}cpu_i}$.

从上述定义可以看出,在虚拟机处理速度相同的情况下,虚拟机上服务请求的总计算量越大,虚拟机的负载率就越大.根据能力与负载相匹配原则,同种类型的虚拟机之间达到负载平衡时应有:

$$\frac{L_{1j}}{w_{1j}^{cpu}cpu_1} = \frac{L_{2j}}{w_{2j}^{cpu}cpu_2} = \dots = \frac{L_{nj}}{w_{nj}^{cpu}cpu_n},$$

其中 $j=1,2,\dots,m$ 为虚拟机类型的数目.云虚拟机中心第 j 类型虚拟机的平均负载率 \overline{lr}_{ij} 为 $\overline{lr}_{ij} = \frac{1}{n} \sum_{i=1}^n \frac{L_{ij}}{w_{ij}^{cpu}cpu_i}$.

(2) 服务请求迁移的目标虚拟机的选取

在云虚拟机中心,虚拟机负载信息的收集、传播以及服务请求真正到达目标虚拟机之间通常具有时间差,因此,在做出负载均衡决策时所使用的负载信息往往都是过时的信息^[20],过时的信息可能导致错误的决策.本文在解决信息过时的 k 子集算法^[20]的基础上,采用在所有虚拟机中挑选 k 个负载最少的虚拟机,然后从中随机选择一个虚拟机作为服务请求迁移的目标虚拟机,既能保证选择到负载相对较少的虚拟机,又能提高选择的效率,可有效解决群聚效应的发生以及轻载虚拟机立即成为重载虚拟机的问题.实验结果表明, k 的取值一般为 $\lceil \log_2 n \rceil$,其中 n 为系统中同类型虚拟机的个数.

(3) 云虚拟机中心同类型虚拟机间全局负载均衡方法

在设计一个全局负载均衡策略时,有 3 个问题需要我们来考虑.(1) 什么时候需要在虚拟机中心的同类虚拟机间进行负载均衡,两次相邻的负载均衡的时间间隔应该设置多长.如果负载均衡的时间间隔太长,则不能及时响应虚拟机上的服务请求对资源增长的需求.相反,如果时间间隔太短,则整个虚拟机中心将由于频繁的服务请求迁移而处于震荡状态,虚拟机处理性能也会受影响因而下降.本文根据云中心物理机资源利用率的情况来决定是否进行负载均衡,即如果处于繁忙状态的物理机数目超过设定的阈值,进行负载均衡,阈值的大小反映了负载均衡的频度.(2) 哪些虚拟机上的服务请求应该被迁移,应该迁往哪些虚拟机上.利用门限策略在虚拟机之间进行负载迁移,设定虚拟机负载迁出的发送门限值和虚拟机接收负载的接收门限值,发送门限值大于接收门限值.当虚拟机的负载值大于发送门限值时,该虚拟机将被确定为负载发送虚拟机;当虚拟机的负载值小于接收门限值时,该虚拟机将被确定为负载接收虚拟机.(3) 服务请求迁移的数量.根据虚拟机应分配与之能力相匹配的负载的原则,对高于平均负载率的虚拟机,减少其上的服务请求数量;对低于平均负载率的虚拟机,增加分配给它的服务请求数量.根据上述原则,负载均衡后虚拟机 vm_{ij} 的服务请求数 \widehat{sn}_{ij} 应为 $\widehat{sn}_{ij} \left[w_{ij}^{cpu}cpu_i \times \sum_{k=1}^n \frac{sn_{kj}}{w_{kj}^{cpu}cpu_k} \right] / n$.

从上述负载均衡过程可以看出,若某类应用的总体服务请求量较大,通过执行该类应用的虚拟机间的负载均衡,使得该类应用的虚拟机都呈现出较高的负载率,进而促进每个物理机分配更多的物理资源给该类虚拟机.

5.3 云虚拟机中心基于灰色波形预测的按需资源分配算法

本文中,为虚拟机按需分配 CPU、内存和网络通信资源的资源分配模式是相同的,我们仅就 CPU 资源的分配过程进行详细描述.具体包括两个阶段:(1) 在物理机内,基于对虚拟机负载的灰色波形预测,提前按预测的负载以最大化物理机的服务效用为目标,为虚拟机按需分配物理资源,以尽可能地避免由于各虚拟机负载不均而导致一些虚拟机负载过重、虚拟机的物理资源不足,而另一些虚拟机负载过轻、虚拟机物理资源过剩.(2) 在不同物理机的同类型虚拟机进行全局负载均衡,以实现同类型虚拟机间的按需资源分配.

算法 1. 基于灰色波形预测的按需资源分配算法 ODRGWF.

输入:云计算系统的物理机集合 $PM=\{pm_1, pm_2, \dots, pm_n\}$, 物理机中虚拟机的个数 m , 虚拟机的服务权重集合 $SW=\{sw_1, sw_2, \dots, sw_m\}$.

输出:物理机 pm_i 的每个虚拟机 vm_{ij} 被分配的物理资源数量 r_{ij}^{allo} .

ODRGWF()

```

{   $sw_{ij} \leftarrow sw_j$ ;  $r_{ij}^{allo} \leftarrow r_i/m$ ; //虚拟机等分物理机  $pm_i$  的物理资源
  for each  $T_i$ 
    {  for each  $ts_k \in T_i$  //  $ts_k$  为资源分配周期  $T_i$  的第  $k$  个时间间隔
        for each  $pm_i \in PM$ 
          for each  $vm_{ij} \in pm_i$ 
            {   $d_{ij}^{(k)} \leftarrow record\_coming\_service\_request\_amount(vm_{ij}, ts_k)$ ; //记录到达  $vm_{ij}$  的请求数
                 $d_{ij}^{(k)} \leftarrow record\_finished\_service\_request\_amount(vm_{ij}, ts_k)$ ;
                if ( $r_{ij}^{allo} - r_{ij}^{act} < \varepsilon r_{ij}^{allo}$   $vm_{ij} \in que_{sur}$ )  $que_{sur} \leftarrow que_{sur} - \{vm_{ij}\}$ ;
                else if ( $r_{ij}^{allo} - r_{ij}^{act} \geq \varepsilon r_{ij}^{allo}$ )  $que_{sur} \leftarrow que_{sur} + \{vm_{ij}\}$ ; //  $vm_{ij}$  放入可获取资源队列  $que_{sur}$ 
                if ( $r_{ij}^{act} - r_{ij}^{allo} \geq \delta r_{ij}^{allo}$   $vm_{ij} \in que_{need}$   $\sum r_{ij}^{need} \in que_{need} r_{sur}$ )
                   $vm_{ij} \leftarrow obtain\_resource(r_{ij}^{act} - r_{ij}^{allo}, que_{sur})$ ; //从  $que_{sur}$  为  $vm_{ij}$  分配所需的资源
                else if ( $r_{ij}^{act} - r_{ij}^{allo} \geq \delta r_{ij}^{allo}$   $vm_{ij} \in que_{need}$   $\sum r_{ij}^{need} \in que_{need} > r_{sur}$ )
                   $vm_{ij} \leftarrow obtain\_resource(r_{ij}^{allo}, r_{tot}) \max \sum_{j=1}^m u_{ij}$ ; /*按最大化各虚拟机的服务效用之和的
                                                                原则为虚拟机重新分配资源*/
                else if ( $r_{ij}^{act} - r_{ij}^{allo} < \delta r_{ij}^{allo}$   $vm_{ij} \in que_{need}$ )  $que_{need} \leftarrow que_{need} - \{vm_{ij}\}$ ;
                else if ( $r_{ij}^{act} - r_{ij}^{allo} \geq \delta r_{ij}^{allo}$ )  $que_{need} \leftarrow que_{need} + \{vm_{ij}\}$ ;
            }
          if ( $T_i$  结束时到来)
            {   $\hat{x}_{ij}^{(0)}(ts_{n+k}) \leftarrow grey\_wave\_forecasting(a_{ij}^{(1)}, \dots, a_{ij}^{(n)}, vm_{ij})$ ; /*用灰色波形预测模型预测下一资源
                                                                分配周期每个时间间隔到达  $vm_{ij}$  的服务请求数量*/
                 $fn_{ij} \leftarrow computePSL(a_{ij}^{(1)}, \dots, a_{ij}^{(n)}, d_{ij}^{(1)}, \dots, d_{ij}^{(n)})$ ; /*计算下一个局部资源再分配周期  $vm_{ij}$  单位时间
                                                                内的服务请求数量的预值  $fn_{im}$  */
                 $r_{ij}^{allo}(T_{i+1}) \leftarrow computeMURA(fn_{i1}, \dots, fn_{im}, sw_{i1}, \dots, sw_{im}, r)$ ; /*根据  $\max \sum_{j=1}^m u_{ij}$  计算  $vm_{ij}$  在下一资源
                                                                分配周期  $T_{i+1}$  的资源分配量  $r_{ij}^{allo}(k)$  */
                Adjusting quantity of resource for  $vm_{ij}$  according to  $r_{ij}^{allo}(T_{i+1})$ ; ;
            }
        }
    }
  if ( $busy\_processors\_amount(PM) > \lambda n$ ) //繁忙的物理机数量大于设定的阈值,进行全局负载均衡
    {  for( $j=1$ ;  $j \leq m$ ;  $j++$ )
        {  for each  $vm_{ij}$  //对同种类型的虚拟机进行负载均衡
            if ( $lr_{ij} < \text{接收门限值}$ )  $G_{receive} \leftarrow G_{receive} + \{vm_{ij}\}$ ; //  $vm_{ij}$  进入负载接收虚拟机队列  $G_{receive}$ 
            else if ( $lr_{ij} > \text{发送门限值}$ )  $G_{send} \leftarrow G_{send} + \{vm_{ij}\}$ ; //  $vm_{ij}$  进入负载发送虚拟机队列  $G_{send}$ 
            do until 第  $j$  类型的虚拟机之间负载均衡
              {   $x \leftarrow |G_{receive}|$ ; //得到  $G_{receive}$  中虚拟机的个数
                   $G \leftarrow select\_min\_load(vm_{ij}, G_{receive}, [\log_2 x])$ ; //选择  $[\log_2 x]$  个负载最小的虚拟机
                   $vm_{uj} \leftarrow randomly\_select\_vm(G)$ ; //在  $G$  中随机选择一个虚拟机
              }
            }
        }
    }

```

```

 $G_{receive} \leftarrow G_{receive} - \{vm_{ij}\}; vm_{vj} \leftarrow randomly\_select\_vm(G_{send}); G_{send} \leftarrow G_{send} - \{vm_{vj}\};$ 
 $load\_balancing(vm_{uj}, vm_{vj}, \bar{lr}_{ij});$  //在  $vm_{uj}$  和  $vm_{vj}$  之间按平均负载率  $\bar{lr}_{ij}$  进行负载均衡
if (负载均衡后的  $vm_{uj}$  < 接收门限值)  $G_{receive} \leftarrow G_{receive} + \{vm_{uj}\};$ 
else if (负载均衡后的  $vm_{vj}$  > 发送门限值)  $G_{send} \leftarrow G_{send} + \{vm_{vj}\};$ 
} } }
} //end for each  $T_i$ 
} //end ODRGWF()

```

6 模拟实验及结果分析

下面通过模拟实验来测试所提出的虚拟机按需物理资源分配方法的效果.我们用云仿真软件 Cloudsim 3.0^[3]来进行模拟实验,CloudSim 3.0 是基于 Java 的离散事件模拟工具包,支持云计算的资源管理和调度模拟. CloudSim 3.0 提供了以下功能:(1) 支持在一个物理计算节点上对大规模云计算基础设施和数据中心的建模与仿真;(2) 一个建模数据中心、服务代理、调度和分配策略的独立平台;(3) 在一个数据中心节点内为服务生成和管理虚拟机.CloudSim 3.0 模拟程序运行在 Intel 奔腾双核 E5800,3.2GHz,2GB DDR3,Windows XP 专业版 32 位 SP3 操作系统的戴尔台式机上.

6.1 物理机内按需资源再配置策略的效果

本实验验证物理机内基于灰色波形预测为各虚拟机按需分配 CPU 资源后,各虚拟机 CPU 资源实际利用率的情况.选择 XEN4.1 虚拟机作为虚拟服务器,选择 Intel 奔腾双核 E5800,3.2GHz,2GB DDR3,Windows XP 专业版 32 位 SP3 操作系统的戴尔台式机作为物理机实验平台.在 Guest OS 中运行 CPU 负载,GUEST OS 配置如下:512MB RAM,1 个 VCPU.实验负载按 CPU 使用量进行模拟,服务请求到达服从指数分布.物理机内设置 3 个虚拟机,初始时为每个虚拟机分配 28% 的物理 CPU,设定 3 种服务请求的计算量相同,3 个虚拟机的服务权重相同,时间间隔为 30s.随着时间的推移,基于灰色波形预测为虚拟机按需分配资源的分配效果分别如图 3(a)~图 3(c) 所示.

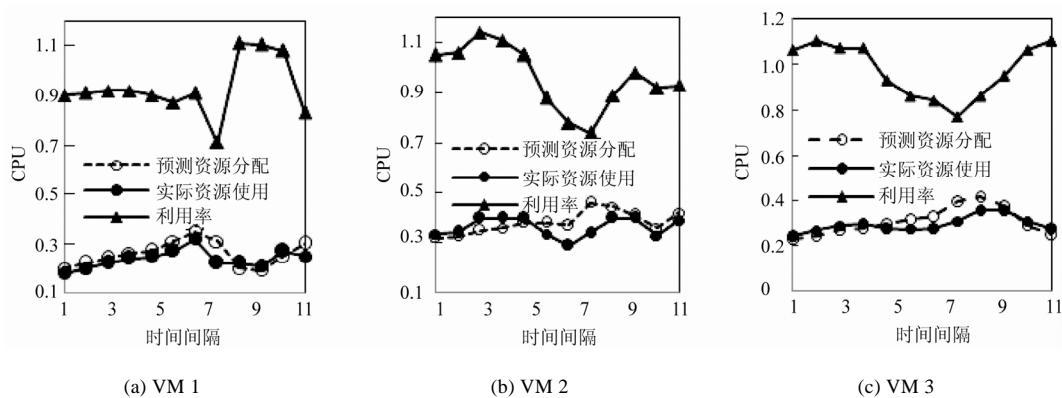


Fig.3 Effect of the allocation of resources to VMs on-demand

图 3 虚拟机按需分配资源的分配效果

从图 3 中 VM 1, VM 2 和 VM 3 的资源使用情况可以看出,基于灰色波形预测为虚拟机按需分配的资源量与虚拟机实际资源使用量非常接近,尽管在负载变化较大的时间段预测误差较大,但也保持了较好的接近,利用率分别达到 71%, 74% 和 77%;当负载变化平稳时,基于灰色波形预测分配资源的效果更好,资源利用率在 90% 上下波动.当图中资源利用率大于 1 时,表示资源分配不足.从 3 个虚拟机的资源利用率变化曲线中可以看出,在负载变化不大时,随着虚拟机的负载趋于平稳,3 个虚拟机都保持了较高的资源利用率,虚拟机的资源利用率收敛于

一个较高的水平.在 VM 1,VM 2 和 VM 3 内,预测的资源分配量与实际的资源使用量之间的平均误差分别为 11.9%,10.8% 和 10.4%,这说明我们所采用的灰色波形预测模型的预测准确度是相当高的.

6.2 ODRBLB负载均衡效果实验

本实验验证使用 ODRBLB 负载均衡策略与不使用该策略时,系统中虚拟机资源利用率的情况.实验参数设置如下:使用 150 个物理机,处理速度分别为 1 000MIPS,2 000MIPS 或 3 000MIPS;每个物理机内部署 3 个不同类型的虚拟机,3 种虚拟机的服务权重相同.设定当虚拟机资源利用率 75%时,虚拟机处于超载状态;当虚拟机资源利用率 15%时,虚拟机处于轻载状态.

随着服务请求数量的增加,系统中超载虚拟机、轻载虚拟机的数量变化情况如图 4 所示.

从图 4 可以看出,随着服务请求数的增加,系统中轻载虚拟机的个数都逐渐减少,重载虚拟机的个数都逐渐增加.运行 ODRBLB 的云系统,其轻载虚拟机的个数远小于未运行 ODRBLB 的云系统的轻载虚拟机个数,超载虚拟机的个数也小于未运行 ODRBLB 的云系统的超载虚拟机个数.之所以出现上述情况,是因为 ODRBLB 出于负载均衡,从重载虚拟机上迁移了部分服务请求到轻载虚拟机,既减少了重载虚拟机的个数,也减少了轻载虚拟机的个数.未运行 ODRBLB 的云系统,随着任务数的增加,其轻载虚拟机个数下降的幅度远小于运行 ODRBLB 的云系统的轻载虚拟机个数下降的幅度,相对而言,这说明在运行过程中将存在更多的虚拟机处于空闲状态,造成更多的物理资源浪费.

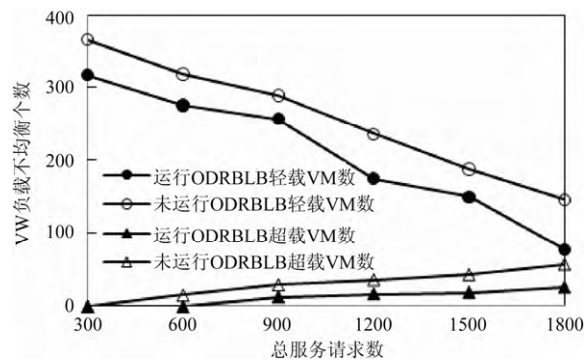


Fig.4 Number of VMs in overload or light-load

图 4 超载或轻载的 VM 个数

6.3 ODRIP,ODRBLB和ODRGWF虚拟机按需资源再配置效果实验

本实验验证分别用物理机内基于灰色波形预测的虚拟机按需资源再配置策略 ODRIP、基于全局负载均衡的虚拟机按需资源再配置策略 ODRBLB 以及二者相结合的基于灰色波形预测的按需资源再配置策略 ODRGWF 为虚拟机进行物理资源分配的效果.实验参数设置如下:使用 200 个物理机,物理机的计算速度为 300 单位计算量/秒;每个物理机内设置 3 个虚拟机,每个虚拟机用来执行一类服务请求;3 种服务请求的计算量都为 4 个单位;3 个虚拟机的服务权重相同;另由 3 个物理机分别按指数分布向 3 类虚拟机发送服务请求;局部资源再分配的周期是 40s,每个周期分成 4 个等时间间隔.实验开始时,每个虚拟机各分配物理机 CPU 计算资源的 1/3.每种请求速率的实验执行 5 个周期,执行 5 次,实验结果取其平均值.

ODRIP 只进行物理机内各虚拟机的按需物理资源再分配,而 ODRBLB 只用服务请求的迁移来缓解虚拟机物理资源的不足.ODRGWF 通过物理机内各虚拟机的按需物理资源再分配和服务请求在同类型虚拟机间的迁移来共同缓解虚拟机对物理资源的竞争.图 5 显示了这 3 种策略为虚拟机分配物理资源的效果.

从图 5 可以看出,当总请求速率达到 500 个/秒时,3 种策略的服务请求完成率都开始下降,但 ODRGWF 的请求完成率最高,ODRIP 的请求完成率最低.这是因为,在物理机内,3 个虚拟机共享物理机的物理资源,当物理机内

的3个虚拟机对物理资源的需求超出了物理机的资源总量时,按 ODRIP 策略为物理机内的虚拟机按需再配置资源,该物理机就无法满足这些虚拟机对资源的需求,而基于全局负载均衡的虚拟机按需资源再配置策略 ODRBLB 通过服务请求的迁移,可缓解重载物理机物理资源的不足,使轻载物理机的物理资源得到更好的利用,因而 ODRBLB 的服务请求完成率高于 ODRIP 的服务请求完成率.ODRGWF 在物理机内部根据虚拟机负载的变化动态按需调整分配给虚拟机的物理资源量,同时通过服务请求迁移缓解单个物理机物理资源的不足.该策略既能及时响应虚拟机对物理资源的需求,又能缓解重载物理机物理资源的不足,因而 ODRGWF 的请求完成率最高.由此可见,ODRIP 和 ODRBLB 这两个策略互为补充,并且缺一不可.

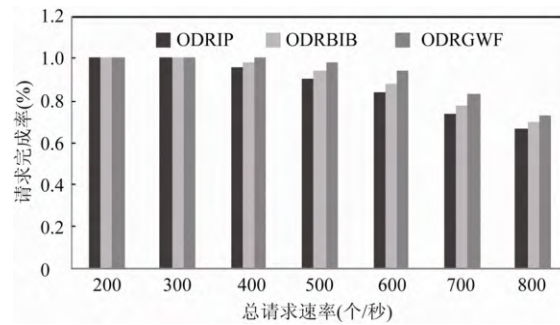


Fig.5 Effect comparison among ODRIP, ODRBLB and ODRGWF strategies

图 5 ODRIP,ODRBLB 和 ODRGWF 这 3 种策略的效果比较

6.4 虚拟机中心不同类型虚拟机的平均请求完成率实验

在云虚拟机中心,为了验证本文所提出的基于灰色波形预测的最大化物理机服务效用的按需资源分配算法 ODRGWF 的性能,将其与文献[9]所提出的按需分配算法(记为 TTODA)和等额资源分配算法(ERAA)进行对比.实验参数设置如下:物理机 pm_i 的平均计算速率为 300 单位计算量/秒,物理机总数为 300,每个物理机内部署 3 个不同类型的虚拟机;采用 CPU 密集型(记为 C)、内存密集型(记为 M)、网络密集型(记为 N)这 3 种服务请求,3 种服务请求的计算量分别为 2,3 和 4;由 3 个物理机分别按指数分布向 3 类虚拟机发送服务请求;3 种虚拟机的服务权重分别为 0.45,0.3 和 0.25.虚拟机中心 3 类虚拟机在不同资源分配算法下的平均请求完成率如图 6(a)~图 6(c)所示.

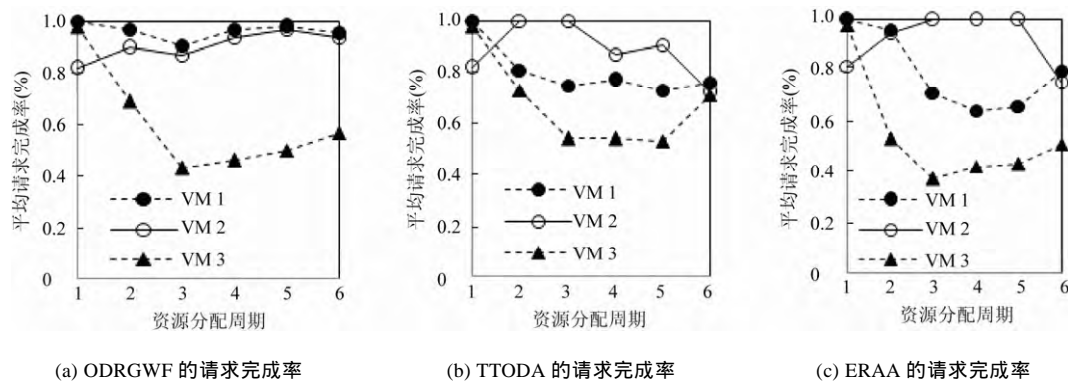


Fig.6 Effect comparison among GFMSU, TTODA and ERAA resource allocation algorithms

图 6 GFMSU,TTODA 和 ERAA 这 3 种资源分配方法的效果比较

从图 6(a)~图 6(c)可以看出,在各个资源分配周期,对服务权重较大的虚拟机 VM 1 和 VM 2 来说,ODRGWF 资源分配算法都使它们保持了较高的服务请求完成率.这是因为,TTODA 用最近时间间隔内到达的服务请求量

作为将来负载的预测,若按该预测负载为虚拟机按需分配物理资源,在虚拟机的负载将来呈现上升趋势时,易造成资源分配的不足,反之,造成部分资源浪费.ODRGWF 较为准确地预测到下一资源分配周期的负载,按其预测值按需为虚拟机分配物理资源能够较好地满足虚拟机上的服务请求对处理资源的需求,使得物理机资源在各虚拟机间得到充分利用.ODRGWF 按效用最大化为物理机内的虚拟机分配资源时,服务权值较大的虚拟机能够获得更多的资源,因而服务权重较大的虚拟机保持了较高的服务请求完成率.此外,若某类服务请求的数量较大,通过全局负载均衡可使执行该类服务请求的虚拟机呈现出较高的负载率,促使每个物理机为该类型虚拟机分配更多的物理资源.

6.5 按需为虚拟机分配资源的不同分配方案的效用对比实验

本实验的参数设置如下:物理机的平均计算速率为300单位计算量/秒,物理机总数为500;采用CPU密集型、内存密集型、网络密集型这3种服务请求,3种服务请求的计算量分别为2,3和4;3种虚拟机的服务权重分别为0.5,0.3和0.2.本实验比较ODRGWF和TTODA虚拟机物理资源分配方法所引起的高优先级应用效用的提高、低优先级应用效用的下降情况,实验结果见表1.

Table 1 Utility comparison between ODRGWF and TTODA

表 1 ODRGWF 和 TTODA 的效用对比

	高优先级应用效用提高(%)	低优先级应用效用下降(%)
ODRGWF	26	3
TTODA	28	9

从表1可以看出,TTODA的资源分配方法给高优先级的应用带来了28%的效用提高,但造成低优先级应用9%的效用下降.虽然TTODA的资源分配方法给高优先级应用带来的效用提高(28%)略高于本文ODRGWF资源分配方法的效用提高(26%),但TTODA引起的低优先级应用的效用下降(9%)远高于本文所提出的ODRGWF方法所引起的效用下降(3%).究其原因,本文提出的灰色波形预测模型负载预测方法充分利用了过去负载变化的波形周期性,其预测的负载远比TTODA用刚过去的时间间隔的负载作为虚拟机按需资源分配时的负载更接近真实负载;此外,本文所提出的指数形式的虚拟机服务效用函数可促使系统给高优先级应用的虚拟机分配更多物理资源,从而获得更多的服务效用,但当分配的资源达到一定量时,再分配更多的资源只会带来较少效用的提高,这样可有效抑制过分给高优先级应用的虚拟机分配物理资源,促使将这些资源分配给较低优先级应用的虚拟机,进而获得比将这些资源分配给高优先级应用所得效用更多的服务效用.最后,ODRGWF按处于繁忙状态的物理机数目超过设定的阈值时启动负载均衡比TTODA按固定周期进行负载均衡更有效,更能及时地响应处于重载状态的虚拟机的资源需求.为了监测虚拟机上服务请求数量的变化以及物理机和虚拟机的资源使用状况,虚拟机监测机构和物理机监测机构需要定期发送相关的统计数据到控制资源分配的决策中心,在一个千兆网络中传输这些数据,其开销可忽略.此外,物理机内的虚拟机资源再配置仅会带来0~0.3%的CPU消耗,对系统来说,这些开销也可忽略不计.

7 结束语

本文针对提供单一服务应用的云中心按峰值需求配置处理机资源、服务请求数量随时间动态变化导致资源利用率低下的问题,提出利用虚拟机中心同时提供多种服务应用以提高资源的利用率.采用灰色波形预测算法对下一资源分配周期虚拟机的负载大小进行预测,给出了兼顾资源需求和服务优先等级的虚拟机服务效用函数,以最大化物理机内各虚拟机的整体服务效用值为目标,动态地为执行不同服务应用的虚拟机配置资源,通过全局负载均衡进一步为服务请求量大的虚拟机分配更多的物理资源.最后,模拟实验验证了本文所提出的按需资源分配算法切实可行,对提高云中心的处理机资源利用效率、用户请求完成率以及服务质量都具有实际意义.然而,本文在分配资源时仅仅考虑了CPU或存储资源的单一分配,没有考虑各种资源的联合分配,感知CPU、存储、I/O和通信带宽的需求,并为其分配合适的物理资源是将来要进一步研究的内容.

References:

- [1] Barham P, Dragovic B, Fraser K, Hand S, Harris T, Ho A, Neugebauer R, Pratt I, Warfield A. Xen and the art of virtualization. *ACM SIGOPS Operating Systems Review*, 2003,37(5):164–177. [doi: 10.1145/1165389.945462]
- [2] Matthews JN, Dow EM, Deshanel T. *Running Xen: A hands-on guide to the art of virtualization*. Prentice Hall PTR, 2008.
- [3] Calheiros RN, Ranjan R, Rose CAFD, Buyya R. Cloudsim: A novel framework for modeling and simulation of cloud computing infrastructures and services. *arXiv preprint arXiv:0903.2525*, 2009.
- [4] VMware Infrastructure. Resource management with VMware DRS. VMware Whitepaper, 2006.
- [5] Padala P, Shin KG, Zhu X. Adaptive control of virtualized resources in utility computing environments. *ACM SIGOPS Operating Systems Review*, 2007,41(3):289–302. [doi: 10.1145/1272998.1273026]
- [6] Zhao WM, Wang ZL. Dynamic memory balancing for virtual machines. *ACM SIGOPS Operating Systems Review*, 2009,43(3):37–47. [doi: 10.1145/1618525.1618530]
- [7] Wang XY, Lan DJ, Fang X, Meng YE, Chen Y. A resource management framework for multi-tier service delivery in autonomic virtualized environments. In: *Proc. of the Network Operations and Management Symp.* 2008. 310–316. [doi: 10.1109/NOMS.2008.4575149]
- [8] Song Y, Wang H, Li YQ, Feng BQ, Sun YZ. Multi-Tiered on-demand resource scheduling for VM-based data center. In: *Proc. of the 9th IEEE/ACM Int'l Symp. on Cluster Computing and the Grid*. 2009. 148–155. [doi: 10.1109/CCGRID.2009.11]
- [9] Song Y, Sun YZ, Shi WS. A two-tiered on-demand resource allocation mechanism for VM-based data centers *IEEE Trans. on Services Computing*, 2013,6(1):116–129. [doi: 10.1109/TSC.2011.41]
- [10] Wu H, Zhang WB, Zhang JH, Wei J, Huang T. Benefit-Aware on-demand provisioning approach for virtualresources. *Ruan Jian Xue Bao/Journal of Software*, 2013,24(8):1963–1980 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/4388.htm> [doi: 10.3724/SP.J.1001.2013.04388]
- [11] Mi HB, Wang HM, Yin G, Shi DX, Zhou YF, Yuan L. Resource on-demand reconfiguration method forvirtualized data centers. *Ruan Jian Xue Bao/Journal of Software*, 2011,22(9):2193–2205 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/4056.htm> [doi: 10.3724/SP.J.1001.2011.04056]
- [12] Xu L, Zeng ZB, Yao C. Study on virtual resource allocation optimizationin cloud computing environment. *Journal on Communications*, 2012,33(z1):9–16 (in Chinese with English abstract).
- [13] Wen Y, Meng D, Zhan JF. Adaptive virtualized resource management for application's SLO guarantees. *Ruan Jian Xue Bao/Journal of Software*, 2013,24(2):358–377 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/4216.htm> [doi: 10.3724/SP.J.1001.2013.04216]
- [14] Shi XL, Xu K. Utility maximization model of virtual machine scheduling in cloud environment. *Chinese Journal of Computers*, 2013,36(2):252–262 (in Chinese with English abstract).
- [15] Jin H, Deng L, Wu S, Shi XH, Zhou LK. Automatic power-aware reconfiguration of processor resource in virtualized clusters. *Journal of Computer Research and Development*, 2011,48(7):1123–1133 (in Chinese with English abstract).
- [16] Yang L, Xing HX, Dai Y, Xu S, Wang H, Zhang B. Virtual resource allocation method with the consideration ofperformance interference among virtual machines. *Journal on Communications*, 2014,35(9):79–90 (in Chinese with English abstract).
- [17] Chen Y, Gmach D, Arlit M, Marwah M, Gandhi A. Minimizing data center SLA violations and power consumption via hybrid resource provisioning. In: *Proc. of the 2nd Int'l Green Computing Conf. Piscataway: IEEE*, 2011. 1–8. [doi: 10.1109/IGCC.2011.6008611]
- [18] Montgomery PL. Modular multiplication without trial division. *Mathematics of Computation*, 1985,44(170):519–521. [doi: 10.1090/S0025-5718-1985-0777282-X]
- [19] Lang MX, Fu XY, Zhu GY. *Forecasting Theories and Approaches*. Beijing: Tsinghua University Press, 2011 (in Chinese).
- [20] Mitzenmacher M. How useful is old information? *IEEE Trans. on Parallel and Distributed System*, 2000,11(1):6–20. [doi: 10.1109/71.824633]

附中文参考文献:

- [10] 吴恒,张文博,张建华,魏峻,黄涛.一种收益敏感的虚拟资源按需提供方法. *软件学报*, 2013,24(8):1963–1980. <http://www.jos.org.cn/1000-9825/4388.htm> [doi: 10.3724/SP.J.1001.2013.04388]

- [11] 米海波,王怀民,尹刚,史殿习,周扬帆,袁霖.一种面向虚拟化数字中心资源按需重配置方法.软件学报,2011,22(9):2193-2205.
http://www.jos.org.cn/1000-9825/4056.htm [doi: 10.3724/SP.J.1001.2011.04056]
- [12] 许力,曾智斌,姚川.云计算环境中虚拟资源分配优化策略研究.通信学报,2012,33(z1):9-16.
- [13] 文雨,孟丹,詹剑锋.面向应用服务级目标的虚拟化资源管理.软件学报,2013,24(2):358-377. http://www.jos.org.cn/1000-9825/4216.htm [doi: 10.3724/SP.J.1001.2013.04216]
- [14] 师雪霖,徐恪.云虚拟机资源分配的效用最大化模型.计算机学报,2013,36(2):252-262.
- [15] 金海,邓莉,吴松,石宣化,周理科.一种能耗感知的虚拟集群 CPU 资源自动再配置方法.计算机研究与发展,2011,48(7):1123-1133.
- [16] 杨雷,邢禾续,代钰,徐赛,王昊,张斌.考虑虚拟机间性能互扰的虚拟资源分配方法.通信学报,2014,35(9):79-90.
- [19] 郎茂祥,傅选义,朱广宇.预测理论与方法.北京:清华大学出版社,2011.



曹洁(1980 -),男,河南新乡人,博士,讲师,CCF 学生会员,主要研究领域为并行分布式计算,云计算,可信计算.



曾国荪(1964 -),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为并行分布计算,可信网络软件.



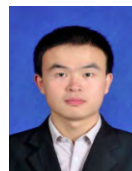
匡桂娟(1972 -),女,博士,CCF 学生会员,主要研究领域为分布式计算,云计算.



张建伟(1971 -),男,博士,教授,主要研究领域为分布并行计算,信息安全.



马海英(1977 -),女,博士,副教授,主要研究领域为信息安全,密码学.



胡克坤(1987 -),男,博士,主要研究领域为并行分布式计算,可靠性.



钮俊(1976 -),男,博士,副教授,CCF 专业会员,主要研究领域为性能评价,模型检测.